

Computación Inteligente: Grandes Modelos de Lenguajes

Clase 7: Agentes LLM

Dr. Wladimir Rodríguez

wladimir@ula.ve

Profesor Titular

Escuela de Ingeniería de Sistemas

ULA

Repaso: Grandes Modelos de Lenguaje

■ ¿Qué son los LLM?

- Modelos de IA entrenados con grandes cantidades de texto
- Uso de arquitecturas de transformers y atención
- Capacidad para generar, interpretar y transformar texto
- Evolución: desde GPT-3 hasta modelos actuales

Repaso: Grandes Modelos de Lenguaje

- **Arquitectura básica de los LLM**
 - Transformers y mecanismos de atención
 - Embeddings y representación vectorial del lenguaje
 - Entrenamiento pre-entrenamiento y fine-tuning
 - Inferencia y generación de texto

Repaso: Grandes Modelos de Lenguaje

- **Capacidades clave de los LLM**
 - Comprensión de contexto y seguimiento de instrucciones
 - Razonamiento y resolución de problemas
 - Generación de contenido en múltiples formatos
 - Zero-shot, few-shot y chain-of-thought reasoning

Limitaciones de los LLM

• Limitaciones fundamentales

- Alucinaciones y generación de información incorrecta
- Razonamiento matemático y lógico inconsistente
- Incapacidad para acceder a información posterior a su entrenamiento
- Dificultades con razonamiento complejo multi-paso

Limitaciones de los LLM

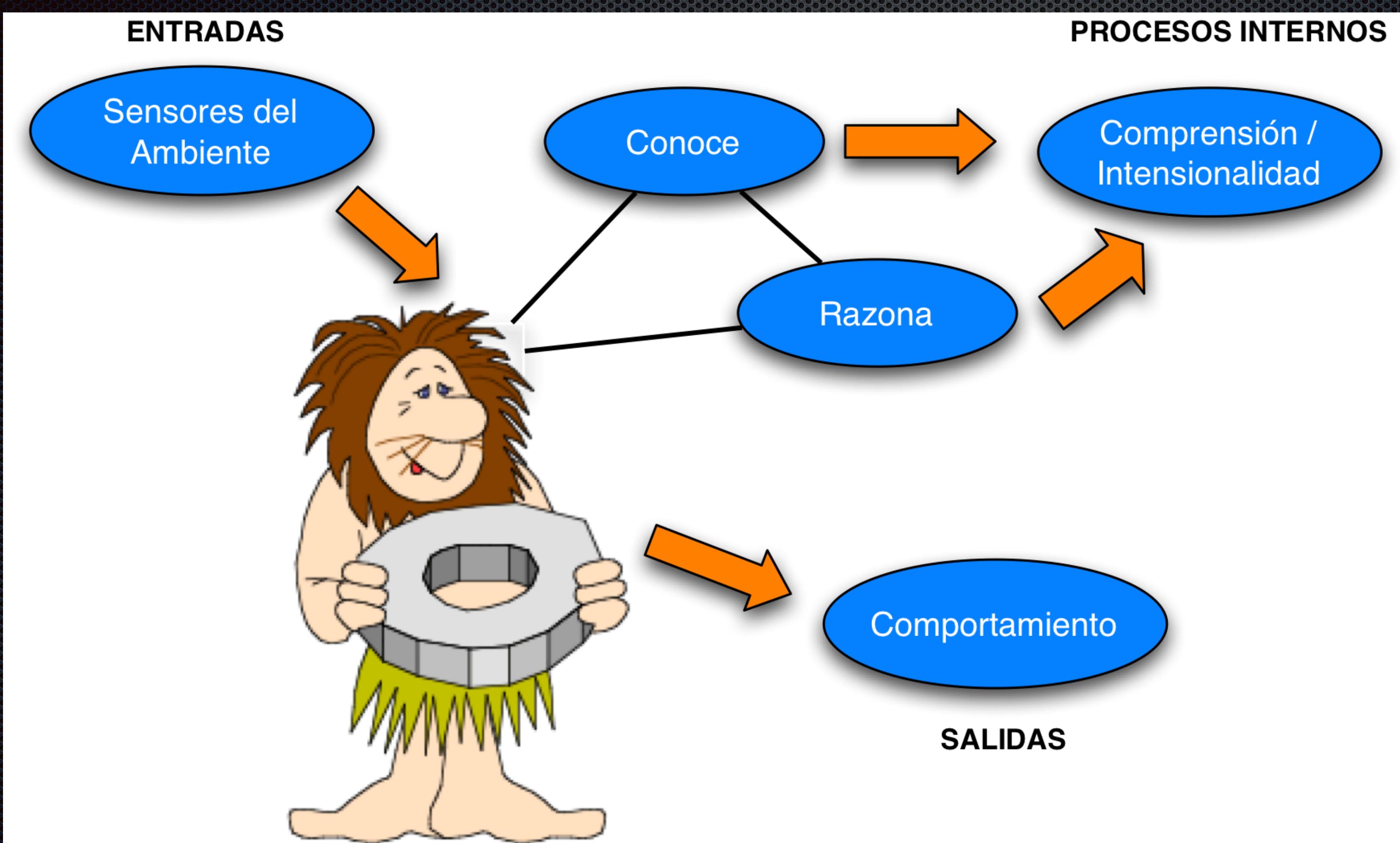
■ Desafíos técnicos

- Límites de contexto (window size)
- Costo computacional de inferencia
- Latencia en generación de respuestas
- Dependencia de la calidad del prompt

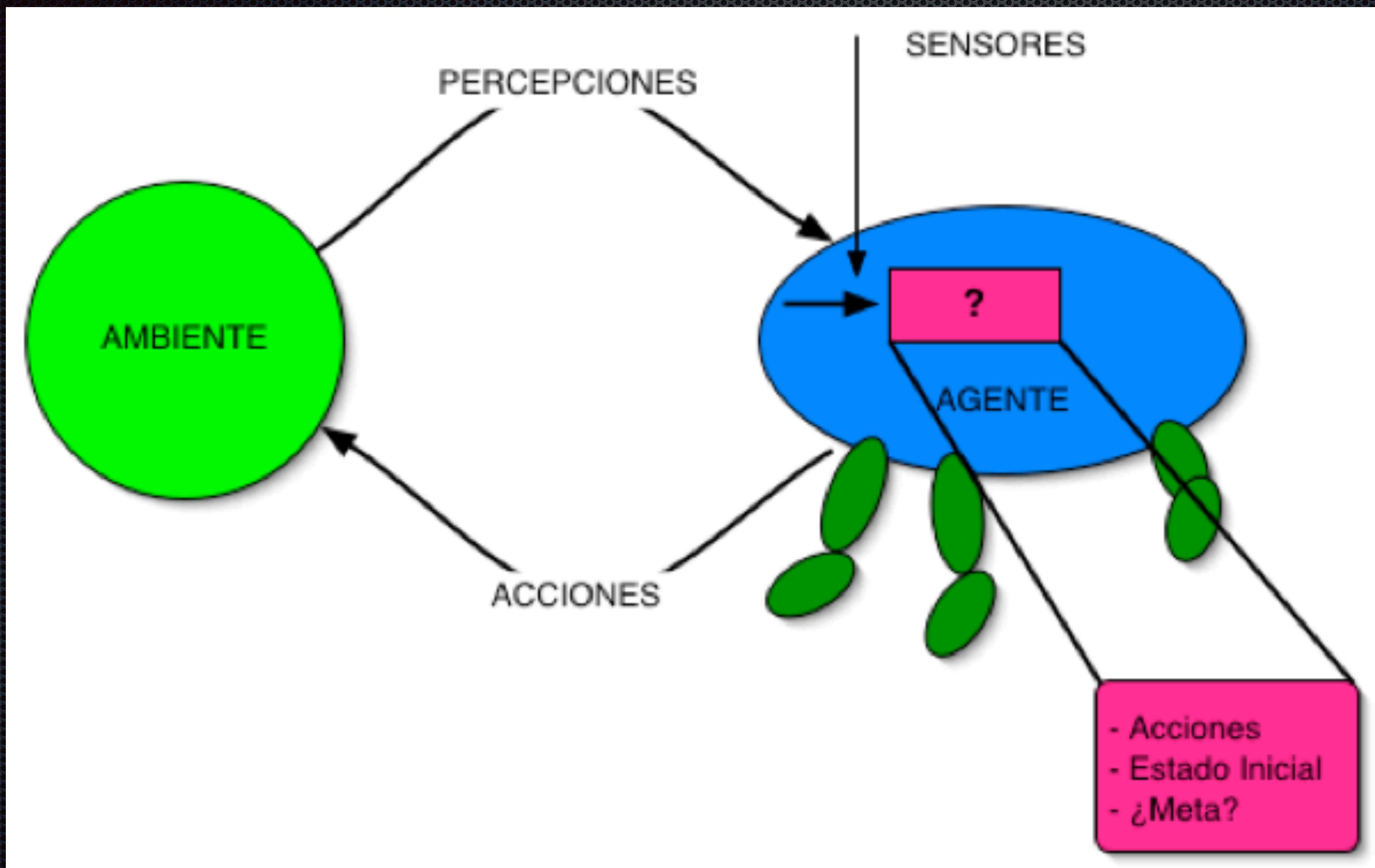
Limitaciones de los LLM

- **Consideraciones éticas**
 - Sesgos aprendidos de los datos de entrenamiento
 - Privacidad de los usuarios y de los datos
 - Problemas de atribución y propiedad intelectual
 - Consideraciones de uso dual (beneficial vs. malicioso)

Agente Humano



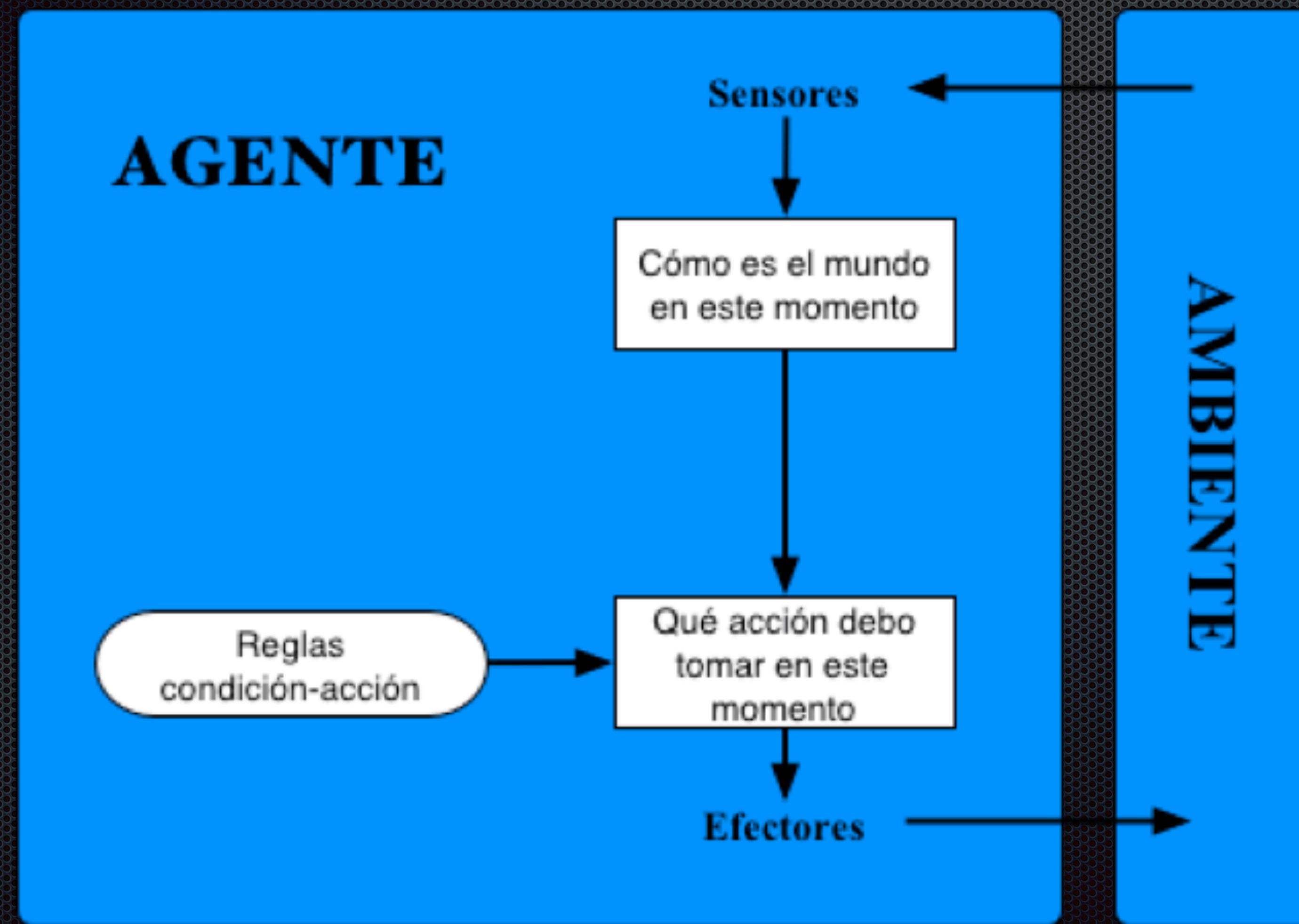
Agente Inteligente



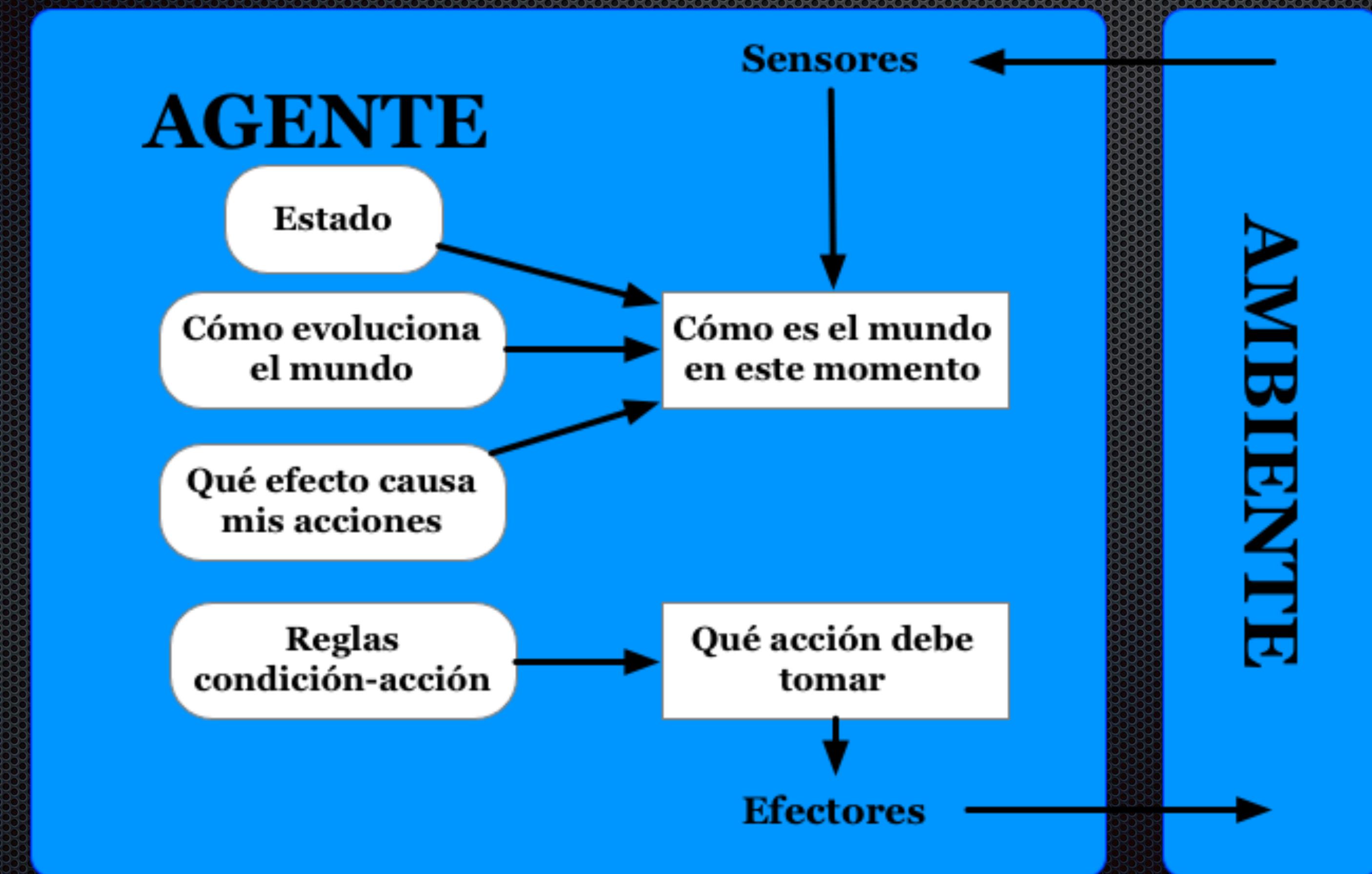
Tipos de Agentes Inteligentes

- Agentes reactivos simples.
- Agentes reactivos basados en modelos.
- Agentes basados en objetivos.
- Agentes basados en utilidad.
- Agentes que aprenden

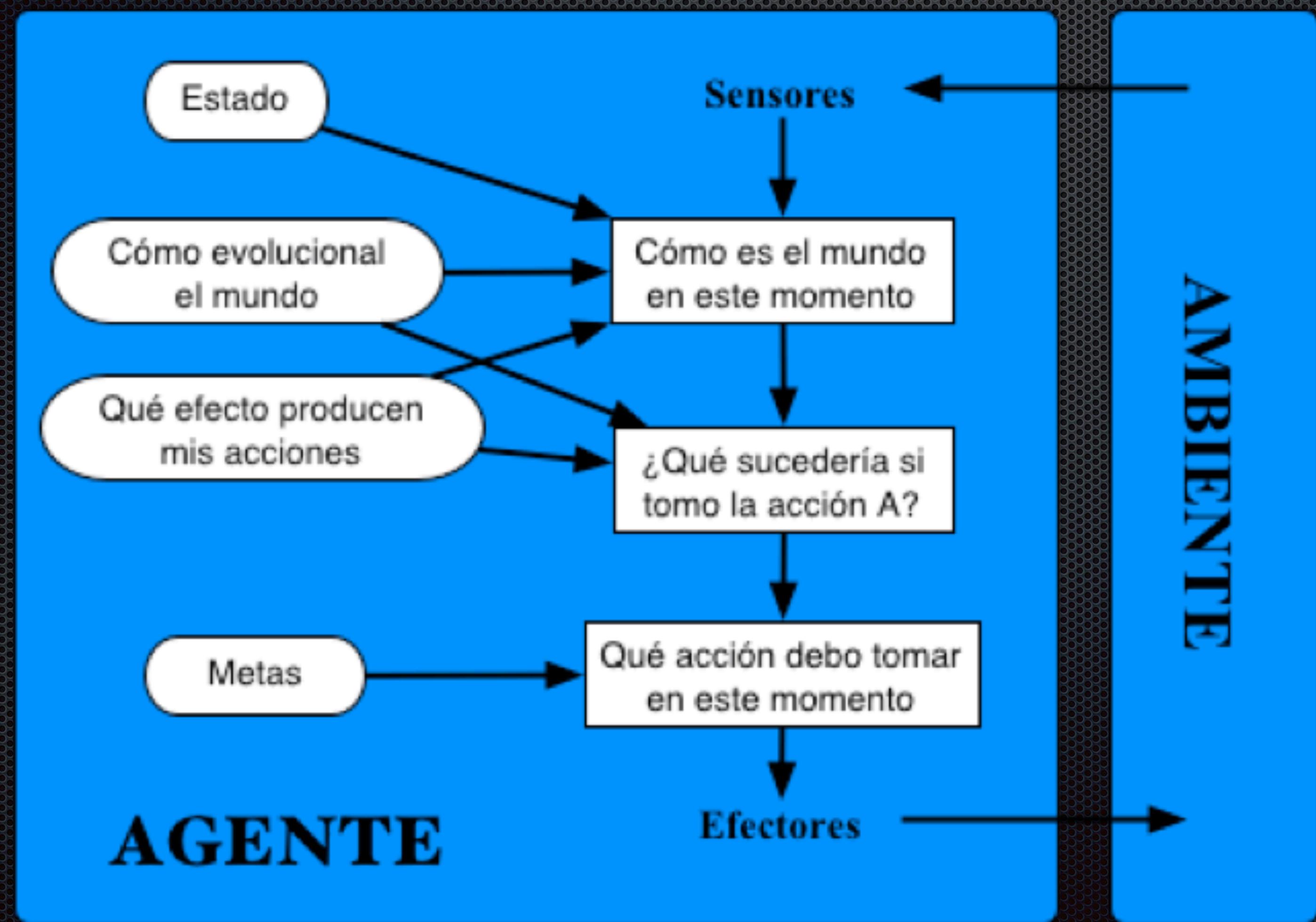
Agentes reactivos simples



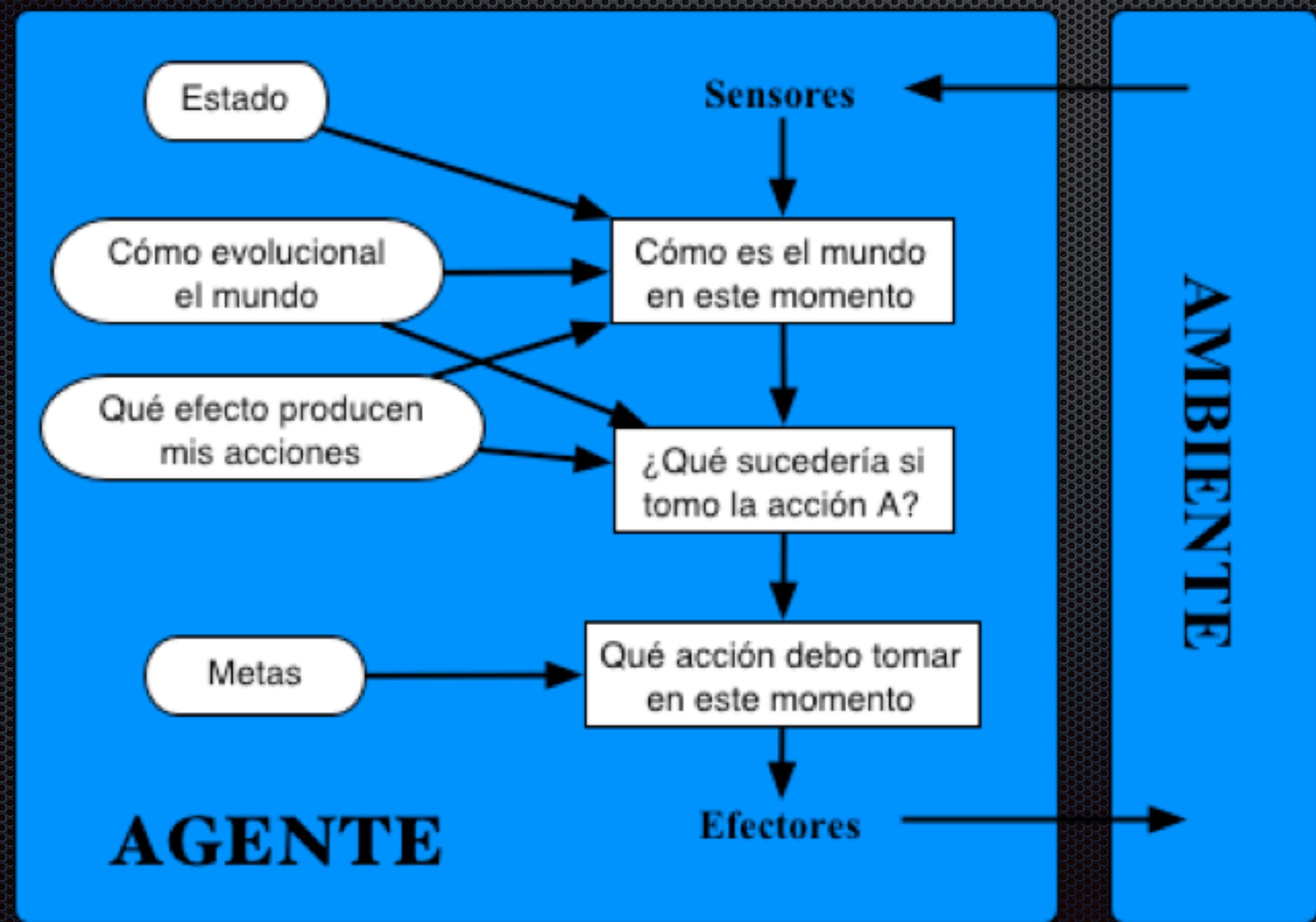
Agentes reactivos basados en modelos



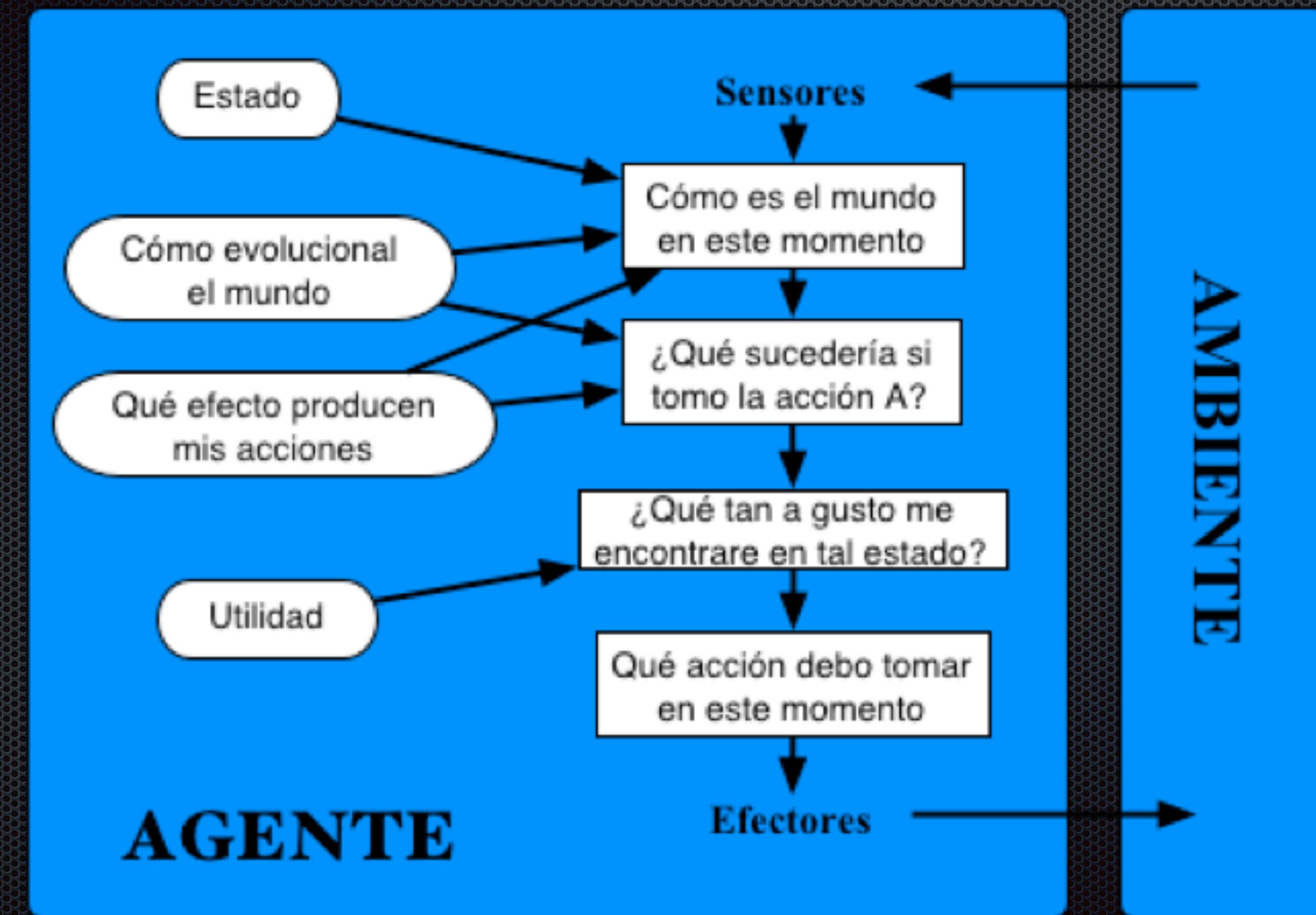
Agentes basados en objetivos



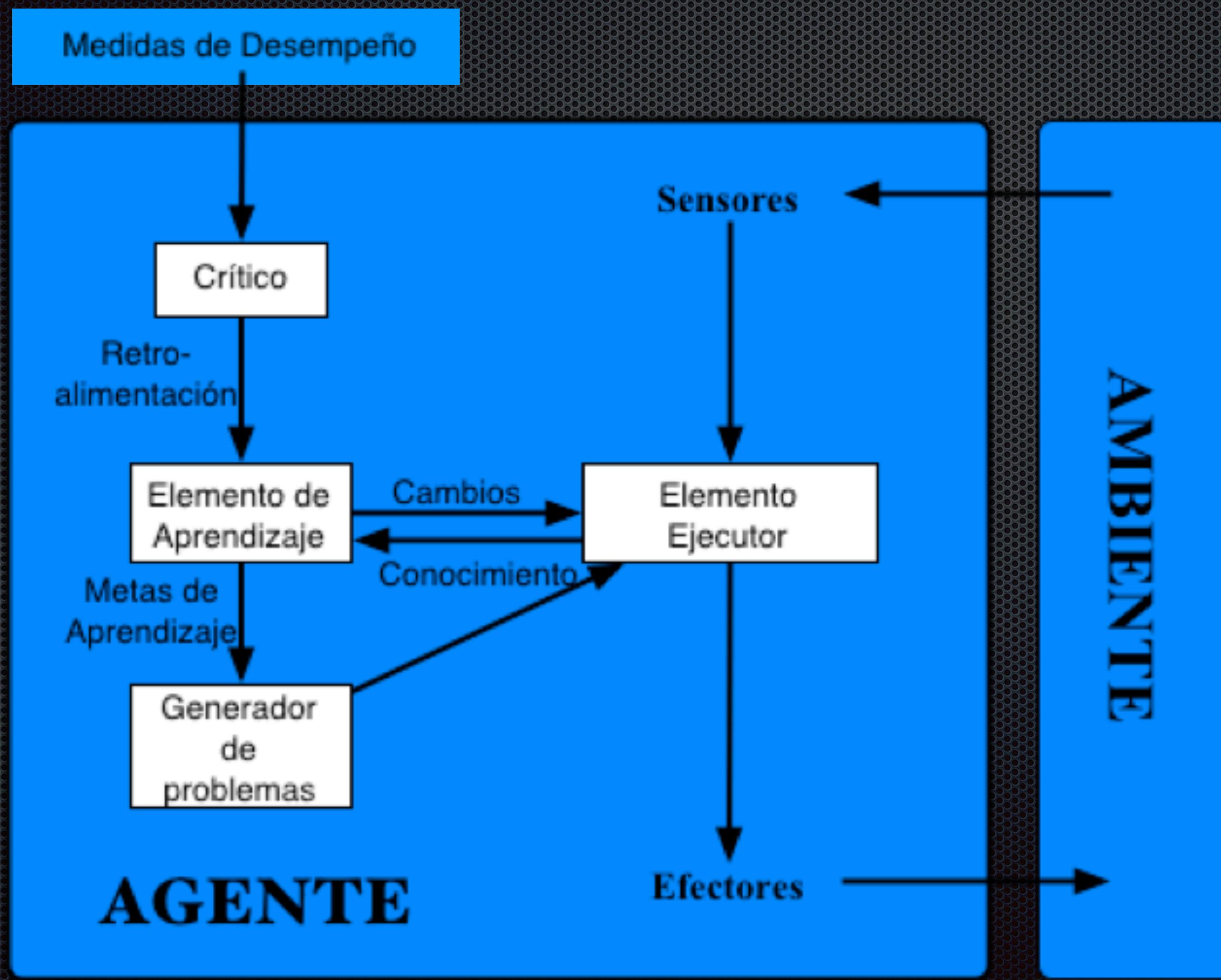
Agentes basados en objetivos



Agentes basados en utilidad



Agentes que aprenden

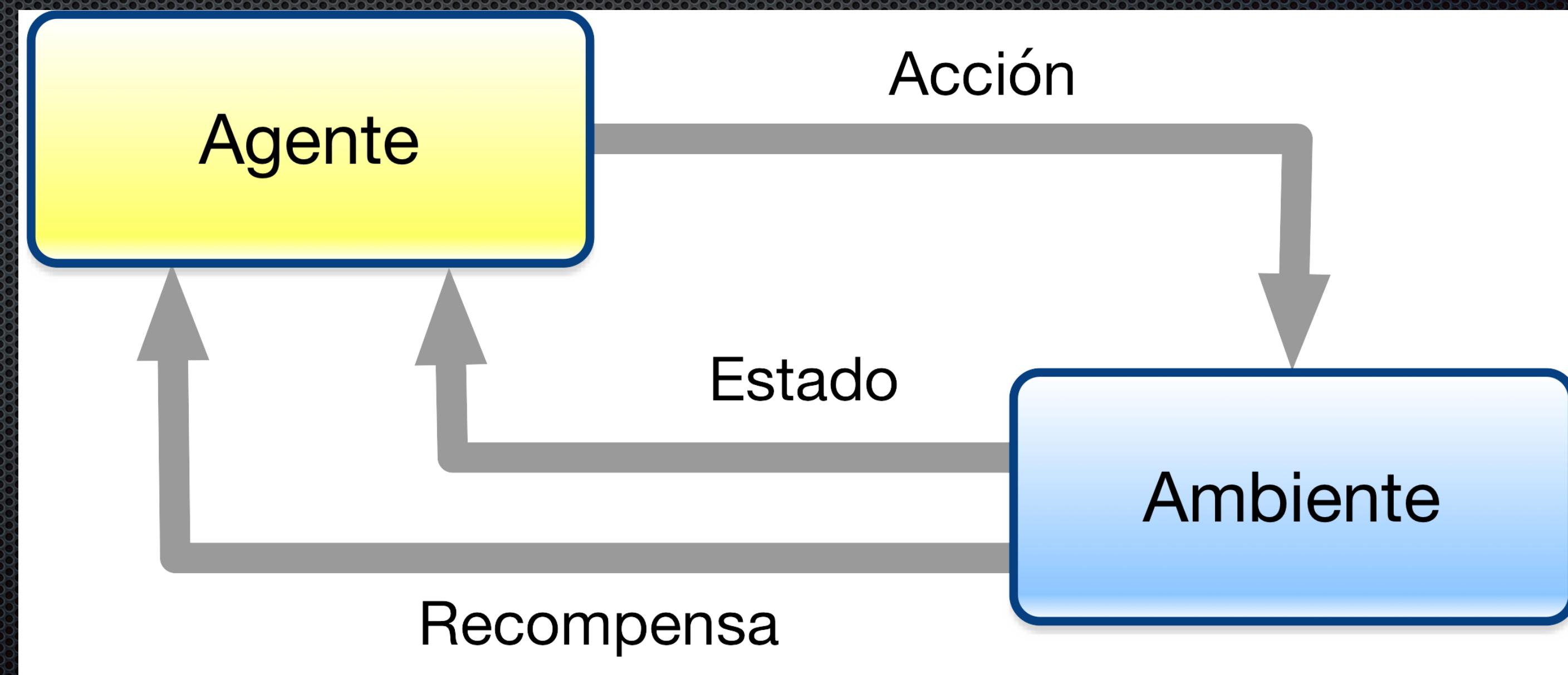


Resumen Agentes Inteligentes

- Un agente es algo que percibe y actúa en un ambiente.
- Un agente ideal es aquel que siempre emprende la mejor acción
- Los agentes de reflejo responden de inmediato a las percepciones
- Los agentes basados en reglas actúan en función del logro de una meta.
- Los agentes basados en la utilidad se esfuerzan por maximizar una función de evaluación.
- El ambiente en el cual se encuentra los agentes pueden variar dramáticamente.

Aprendizaje por Refuerzo

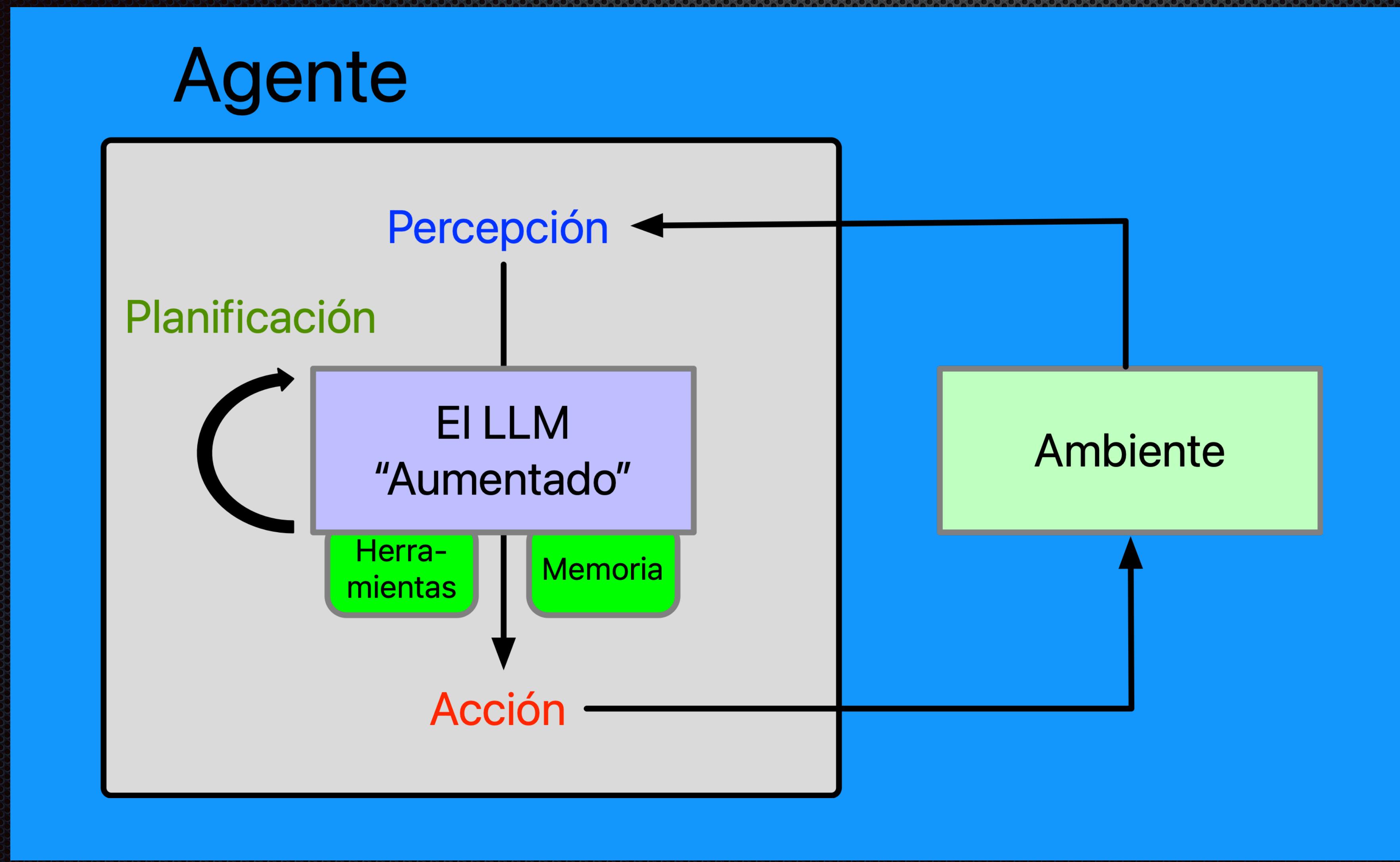
- El aprendizaje por refuerzo es un marco para resolver tareas de control (también llamados problemas de decisión) mediante la creación de agentes que aprenden del ambiente interactuando con él a través de prueba y error y recibiendo recompensas (positivas o negativas) como su única retroalimentación



De LLM a Agentes LLM

- **Definición de agente LLM**
 - Sistema autónomo o semi-autónomo basado en LLM
 - Capacidad para percibir, razonar y actuar
 - Orientación a objetivos y persistencia
 - Integración con sistemas externos

Agentes LLM



De LLM a Agentes LLM

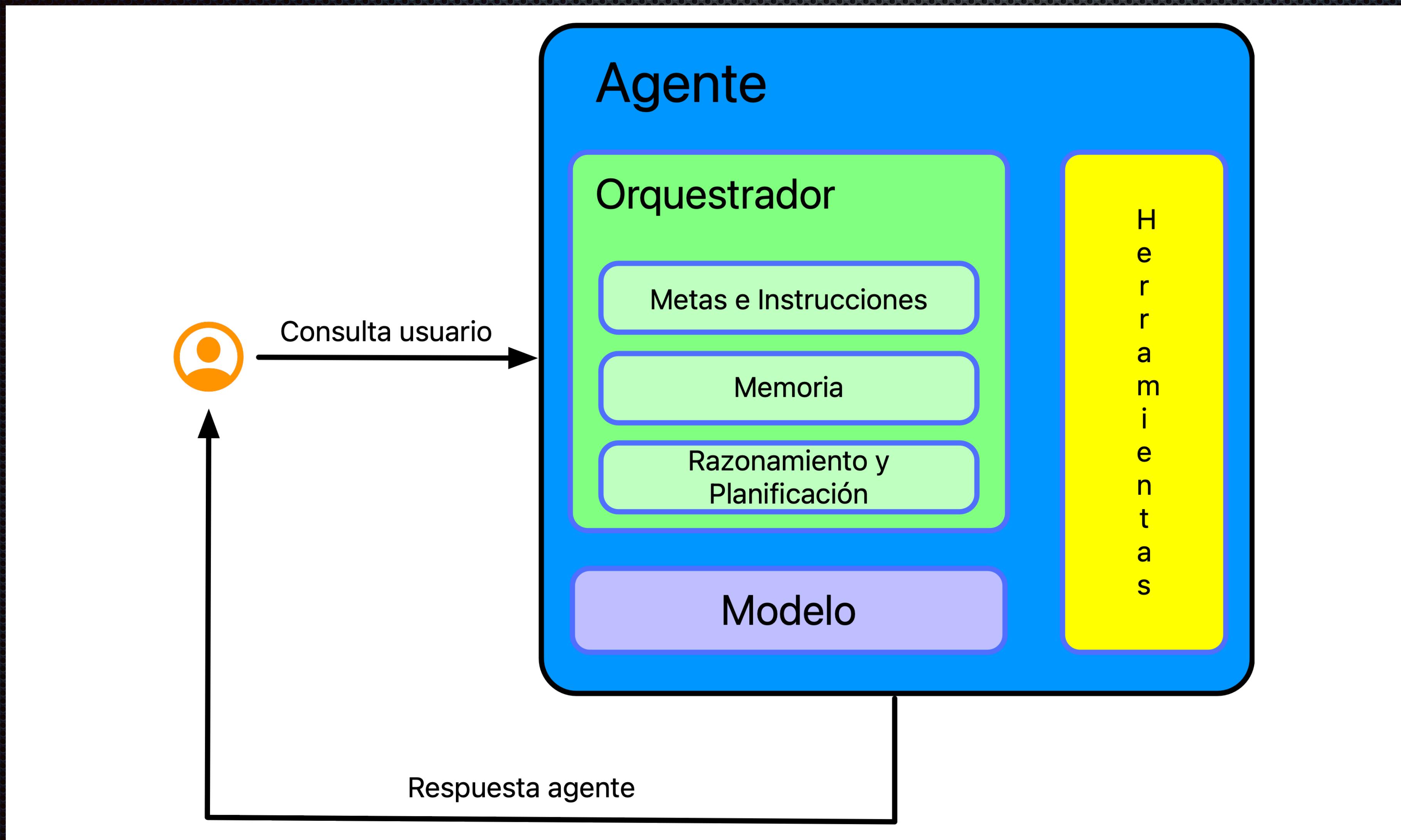
- **Diferencia entre un LLM y un agente LLM**
 - LLM: modelo generativo que responde a estímulos textuales
 - Agente LLM: sistema que utiliza LLM como "cerebro" pero con:
 - Memoria persistente
 - Capacidad de planificación
 - Uso de herramientas externas
 - Retroalimentación y aprendizaje

Arquitectura básica de agentes LLM

- **Componentes fundamentales**

- **Núcleo LLM:** El "cerebro" del agente
- **Sistema de Memoria:** Almacenamiento de información a corto y largo plazo
- **Planificador:** Descomposición de tareas y planificación de acciones
- **Herramientas:** Interfaces con sistemas externos (APIs, bases de datos)
- **Orquestador:** Gestión del ciclo de vida y flujo de ejecución

Arquitectura básica de agentes LLM



Arquitectura básica de agentes LLM

- **Ciclo básico de operación**

- Recepción de instrucción o tarea
- Planificación de acciones requeridas
- Ejecución de acciones (internas o usando herramientas)
- Observación de resultados
- Reflexión y actualización de memoria
- Iteración o finalización

Patrones de diseño para agentes LLM

- **Patrón ReAct (Reasoning + Acting)**
 - Alternancia entre razonamiento y acción
 - Pensamiento paso a paso explícito
 - Uso de observaciones para guiar el siguiente paso
 - Ejemplo práctico de implementación

Patrones de diseño para agentes LLM

■ Patrón Reflexivo

- Auto-evaluación del agente sobre su desempeño
- Capacidad para corregir errores propios
- Refinamiento iterativo de soluciones
- Ejemplos de implementación con LLM

Patrones de diseño para agentes LLM

- **Arquitectura multiagente**
 - Equipo de agentes especializados
 - Comunicación entre agentes
 - Coordinación y distribución de tareas
 - Ventajas y desafíos de implementación

Frameworks actuales para agentes LLM

- LangChain
 - Componentes principales
 - Cadenas y secuencias de ejecución
 - Herramientas y memoria
 - Ventajas y limitaciones

Frameworks actuales para agentes LLM

- AutoGPT y similares
 - Principios de operación
 - Diferencias con otros frameworks
 - Casos de uso recomendados
 - Limitaciones

Frameworks actuales para agentes LLM

- AutoGPT y similares
 - Principios de operación
 - Diferencias con otros frameworks
 - Casos de uso recomendados
 - Limitaciones

Frameworks actuales para agentes LLM

- ❖ Otros frameworks emergentes
 - ❖ CrewAI
 - ❖ Llamalndex
 - ❖ DSPy
 - ❖ Microsoft AutoGen

Comparación con sistemas tradicionales de IA

- **Agentes tradicionales vs. Agentes LLM**
 - Sistemas basados en reglas vs. agentes emergentes
 - Determinismo vs. estocacidad
 - Transparencia vs. opacidad (caja negra)
 - Escalabilidad y adaptabilidad

Comparación con sistemas tradicionales de IA

- **Ventajas competitivas de agentes LLM**
 - Flexibilidad en la interpretación de instrucciones
 - Capacidad para manejar ambigüedad
 - Adaptación a nuevos dominios sin reprogramación
 - Interfaz de lenguaje natural